IETF DPRIVE WG: Encrypting DNS

Sara Dickinson
Sinodun

ICANN 54 - Tech Day
October 2015

DPRIVE WG

Focus is
stub to recursive

Group created

IETF91

IETF92

IETF93

A

ID: problem-statement

RFC7626

ID: dns-tls-newport

ID: start-tls-for-dns

A

ID: start-tls-for-dns

ID:dns-over-tls

STARTTLS & port

Early port allocation

ID: dnsodtls

A

ID: dnsodtls

2014

2015

Q4

Q1

Q2

Q3

Q4

TIME

Pros and Cons

	Pros	Cons
STARTTLS	• Port 53 • Known technique • Incrementation deployment	• Port 53 - middleboxes? • Existing TCP implementations • Downgrade attack on negotiation • Latency from negotiation
TLS (new port)	• New DNS port (no interference with port 53) • Existing implementations	• New port assignment
DTLS	• UDP based • Certain performance aspects	• Truncation of DNS messages (just like UDP) ➡️ Fallback to clear text or TLS ❌ Can't be standalone solution • No running code

Early port allocation

- 8th October 2015 - IANA assigned **port 853**:

domain-s	853	tcp	DNS query-response protocol run over TLS/DTLS
domain-s	853	udp	DNS query-response protocol run over TLS/DTLS

DNS-over-TLS needs TCP !

- DNS-over-TCP… historically used only as a fallback transport (TC=1 ➡️ ‘one-shot’ TCP, Zone transfer)
- 2010: RFC5966 - TCP a **requirement** for DNS implementations
- 2014: Connection-oriented DNS - USC/ISI paper
- draft-ietf-dnsop-5966bis
    - performance on par with UDP, security/robustness
- draft-ietf-dnsop-edns-tcp-keepalive - persistent TCP connections

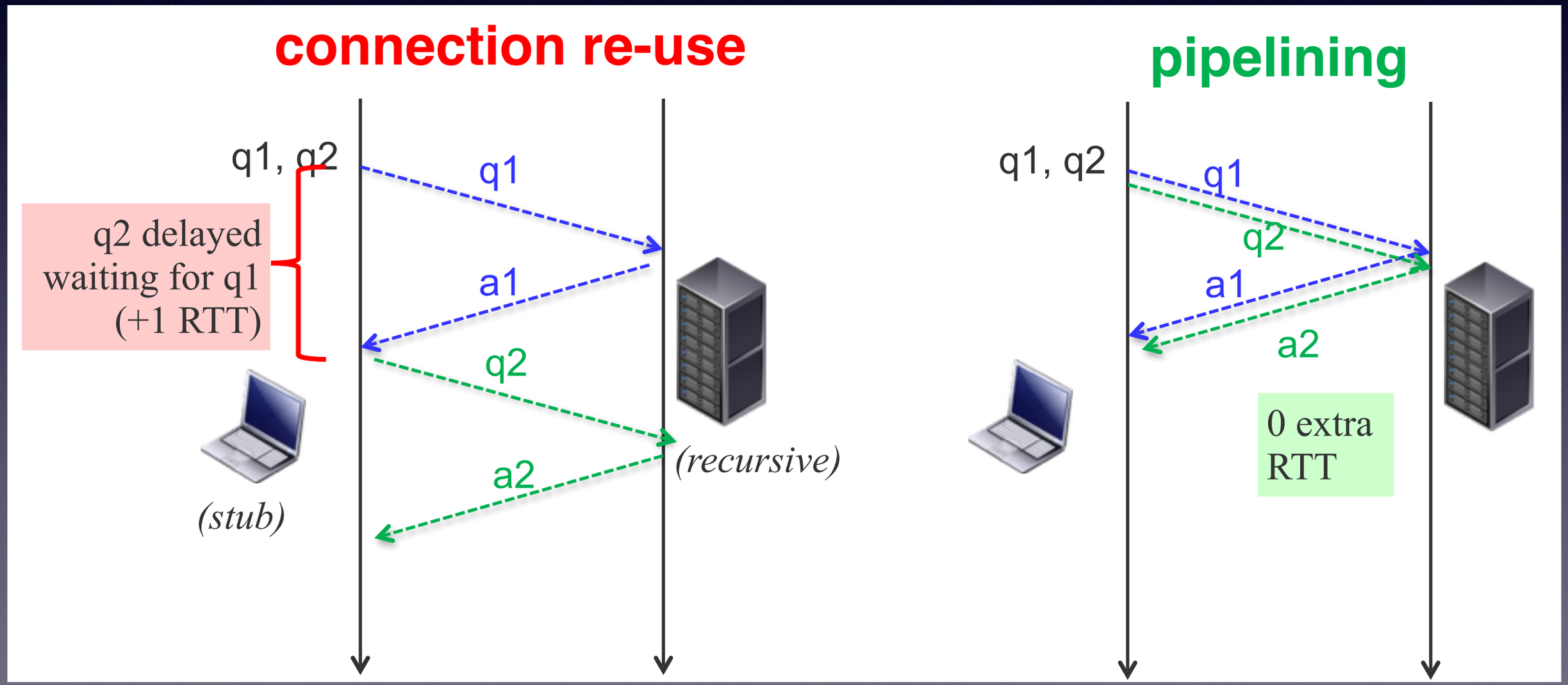
TCP/TLS Performance

Goals:

1. Handle many TCP connections robustly
2. Optimise TCP/TLS set up & resumption
    - TCP FastOpen, TLS resumption, [TLS 1.3]
3. Amortise cost of TCP/TLS setup
    - Send many messages efficiently

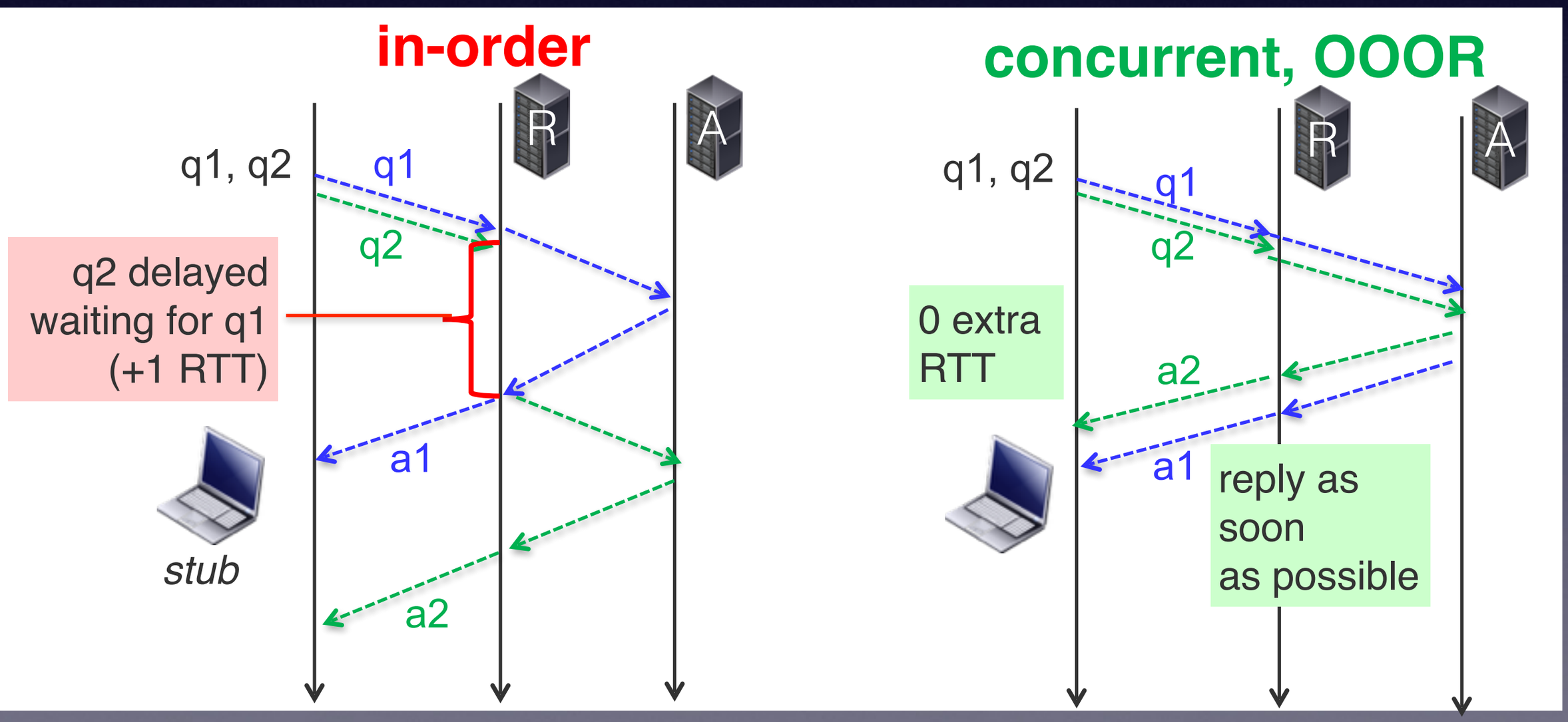
Performance (5966bis)

Client - Query pipelining



Performance (5966bis)

Server - concurrent processing of requests
sending of out of order responses



DNS-over-TLS implementations

- Unbound 1.4.14 (2011) - DNSTrigger
- TLS patches for LDNS and NSD
- [BIND TCP improvements]
- ***getdns*** - ongoing development of DNS-over-TLS

- Modern **async DNSSEC** enabled API
    - https://getdnsapi.net
- Stub mode has TLS with flexible privacy policy and fallback:
    - Strict (Authenticated) TLS only
    - Opportunistic TLS
    - Fallback to TCP, UDP
- Pipelining, OOOP, Configurable idle time

Current status

Software	digit	LDNS	getdns		Unbound		NSD	BIND
mode	client	client (drill)	stub	recursive*	server	client	server	server/ client
TLS								
TFO								
Conn reuse								
Pipelining								
OOOP								

Dark Green: Latest stable release supports this
Light Green: Patch available
Yellow: Patch in progress, or requires building a patched dependancy
Grey: Not applicable or not planned

** getdns uses libunbound in recursive mode*

TLS BCP

- UTA (Using TLS in Applications) WG produced RFC7525 this year - “BCP for TLS and DTLS”

DNS-over-TLS is relatively ‘green-field’

- Key recommendations - Protocol versions:
  - **TLS v1.2** MUST be supported and preferred
- Recommended Cipher Suites (4 of ~100):
  - **AEAD mode** - Forward secrecy for key exchange
  - TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

TLS BCP - Authentication

- Secure discovery of certificate/hostname/etc.
- For DNS-over-TLS?
  - Pre-deployed configuration profile
  - DANE… (clear-text or un-authenticated TLS)
    - boot strap problem

Summary

- Active work on encrypting DNS in DPRIVE
- For DNS-over-TLS performance is key
- Client should consider privacy policy
    - see Appendix for stub/recursive examples
- Know your (D)TLS Best Current Practices

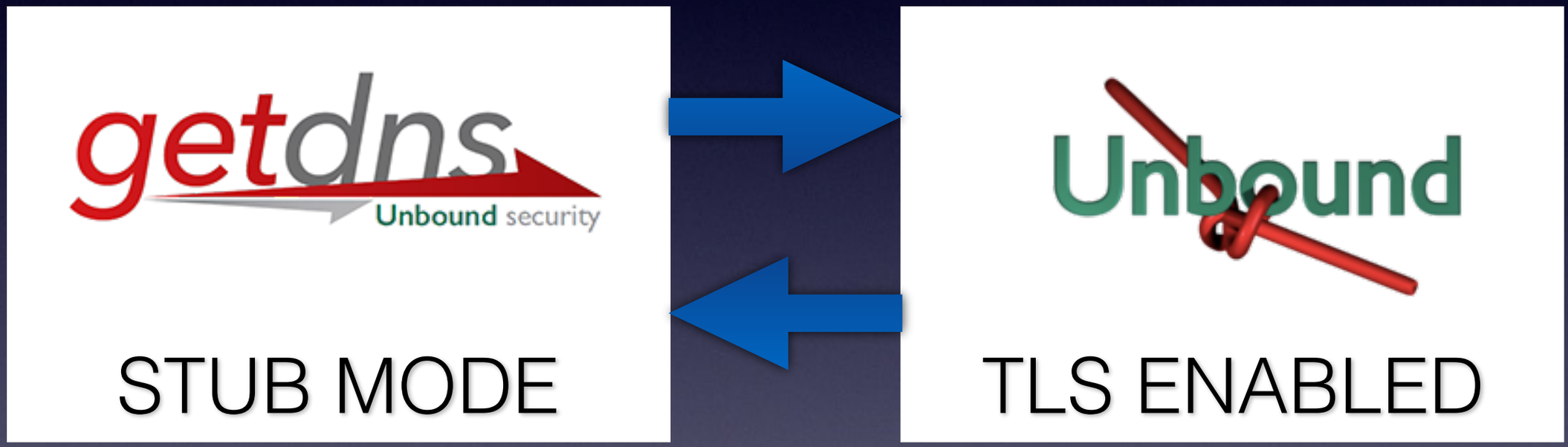
Thank you!

Any Questions?

sara@sinodun.com

Appendix

Examples



Next release:
Hostname verification

1.5.5

Scenario 1:

Strict TLS

- Configuration:
  - **Hostname verification required (Default)**
  - Correct hostname for Unbound resolver
  - TLS as only transport
- RESULT:
  - TLS used (cert & hostname verified)

Scenario 2:

Strict TLS

- Configuration:
  - Hostname verification required (Default)
  - **No or incorrect hostname**
  - TLS as only transport
- RESULT:
  - Query fails

Scenario 3:

Opportunistic TLS

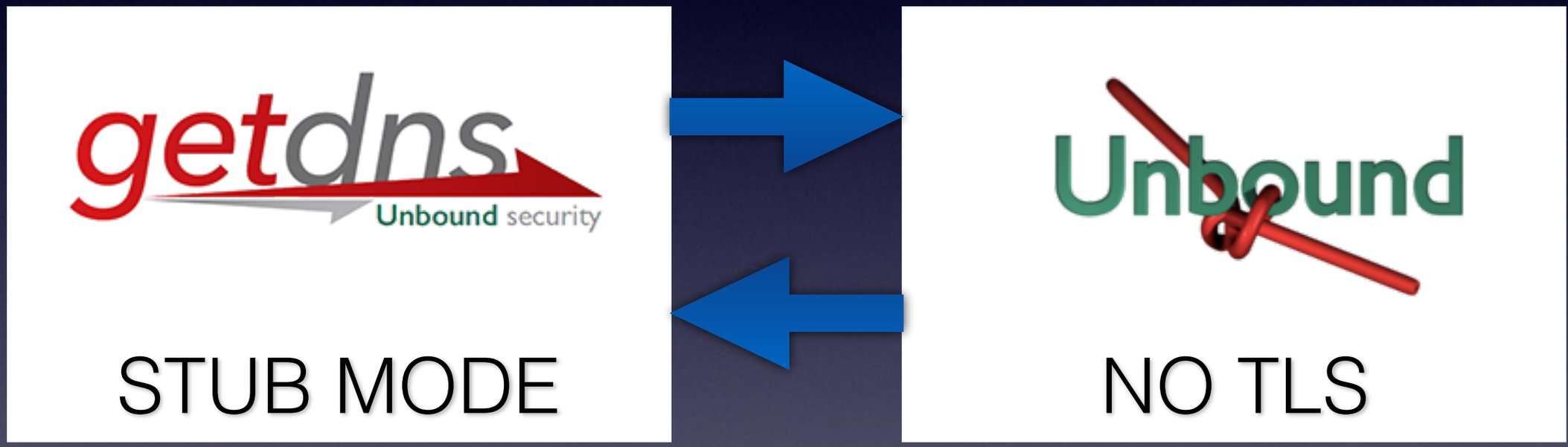
- Configuration:
  - **Hostname verification optional**
  - Valid, none or incorrect hostname
  - TLS as only transport
- RESULT:
  - TLS used (hostname verification tried but fails)

Scenario 4:

Opportunistic TLS

- Configuration:
  - Hostname verification required (default)
  - Valid, none or incorrect hostname
  - **TLS with fallback to TCP**
- RESULT:
  - TLS used (hostname verification tried but fails)

Example



Scenario 3:

Opportunistic TLS

- Configuration:
  - Hostname verification required (default)
  - Valid, none or incorrect hostname
  - TLS with fallback to TCP
- RESULT:
  - TCP used (TLS tried, but fails)